

Description of Cited Art

Waldo describes a distributed system having a lookup service that accommodates adding and deleting services in the lookup service as well as notifying clients when the lookup service has been updated. See Col. 2, lines 50-59. According to Waldo, a service is a resource, data or functionality that can be accessed by a user, program, device or another service. Further, services are logically grouped to form what Waldo calls a "Djinn." Users and devices that join a Djinn may add, delete and access services within the Djinn or be notified when services of the Djinn are registered, deleted or modified. See Col. 4, lines 41-63 and Col. 7, lines 11-19.

When a new device is added to the distributed system a discovery server passes a reference (interface) to the lookup service to the new device. The new device may then use the interface to register its services with the lookup service and become a member of a Djinn. Specifically, a program (client) running on the device adds the device to the distributed system by sending a multicast packet to the discovery server. The discovery server uses a code contained in the packet to send the interface associated with the lookup service to the client. The client utilizes the interface to add, delete or access a service, or request notification when the lookup service is updated. See Col. 5, line 60 to Col. 6, line 3, and Col. 10, lines 47-67.

The interface associated with the lookup service provides various methods including a register method, a lookup method and a notify method. The register method is used by a client to register a new service and to re-register an existing service. The lookup method is used by a client to lookup a particular service. The notify method is

used by a client to register to be notified of an event, such as when the lookup service is updated. See Col. 8, lines 17- 63, Col. 9, lines 1-62 and Col. 11, lines 52-64.

Specifically, a client adds a service by invoking the register method and passing an object or a stub representing the service to the lookup service. A client deletes a service by invoking the lookup service's register method and passing an object or stub representing the service to the lookup service. To access a service, a client invokes the lookup method. The lookup service, in turn, retrieves an object or stub representing the service and passes the object or stub to the client. The client may then access the service by invoking methods using the object or stub. To be notified of updates to the lookup service, a client invokes the lookup service's notify method to register a callback routine that is passed to the lookup service. The lookup service stores information about the client and the callback routine in an event table. See Col. 8, lines 17-63 and Col. 11, lines 1-63.

When an update to the lookup service occurs, the lookup service examines the event table and notifies all clients registered to be notified for the event by invoking each client's registered callback routine. See Col. 12, lines 9-18.

Differences between the Cited Art and the Present Invention

Claim 1 recites:

1. A method for use in a computer network having a process manager and a network management station for reporting to the network management station the addition of new applications or processes to the computer network, the method comprising the steps of:
 - providing a configuration service layer in communicating relationship with a new application or process and the process manager;
 - in response to opening the new application or process, issuing a registration service request from the new application or process to the process manager*** through the configuration service layer;

establishing a method at the network management station for persistently and continuously listening for messages from the process manager;

in response to receiving the registration service request at the process manager, ***generating and forwarding a notification message that identifies the new application or process to the network management station***; and

automatically displaying the notification message at the network management station without having to close and re-start the management station.

Applicant submits that Waldo fails to teach or suggest applicant's claimed "***in response to opening the new application or process, issuing a registration service request from the new application or process to the process manager***" and "***generating and forwarding a notification message that identifies the new application or process to the network management station.***"

According to Waldo, a client "discovers" a lookup service by sending a multicast packet to a discovery server that, in turn, returns an interface to the lookup service to the client. The client may then use the interface to invoke functions provided by the lookup service to add, delete and access services. Nowhere does Waldo teach Applicant's claimed ***in response to opening a new application or process, issuing a registration service request from the new application or process to a process manager***. Rather according to Waldo, a client calls a "ServiceRegistrar" function local to the client to register a service rather than issue a registration service request. See Col. 8, lines 17-30 ("This interface is not a remote interface. Instead, each implementation of the lookup service exports proxy objects that implement the ServiceRegistrar interface local to the client..."). In fact, no where does Waldo teach or suggest issuing a registration request.

Applicant, on the other hand, claims issuing a registration request via a configuration layer to a process manager to register a new application or process with a process manager. Thus, Applicant's claimed invention involves issuing a request to a process manager via a configuration layer that is in communication with the process manager. This is different from Waldo in which registration involves a calling a function local to a client.

With respect to Applicant's claimed "***generating and forwarding a notification message that identifies the new application or process to the network management station***" at best, Waldo teaches invoking a callback function to notify a client that an event has occurred. Thus, in Waldo, notification of an event occurs via the calling of a function not via a notification message as claimed by Applicant. In fact, nowhere does Waldo suggest or teach ***generating and forwarding a notification message*** as claimed by Applicant.

For reasons set forth above, Applicant respectfully urges that Waldo does not render Applicant's claimed invention anticipated under 35 U.S.C. §102.

§103

At paragraph nine of the Office action, claims 8, 13 and 15 were rejected under 35 U.S.C. §103 as being unpatentable over "Monitoring Distributed Systems" by Joyce et al., hereinafter "Joyce," in further view of U.S. Patent 5,655,081 to Bonnell et al., hereinafter "Bonnell."

Description of Cited Art

Joyce describes monitorable processes and channel processes which collect monitoring information from the monitorable processes and distributes this information

to one or more consoles. See pp. 126-127, Section 2.3 and 2.3.1. According to Joyce, when an event is detected in a monitorable process a message containing information about the event is sent to a local channel residing on the same system as the process. See p. 129, Section 2.3.2. A console “plugs” into a channel and collects, interprets and displays the event information reported to the channel. See p. 129-130, Sections 2.3.3 and 2.3.4.

Bonnell describes a technique for monitoring and managing applications and other resources in a computer network in a distributed computing environment. See Col. 6, lines 55-61 and Col. 6, line 67 to Col. 7, line 7. According to Bonnell, a console receives information about events by registering the events with an agent before the events occur. The agent loads knowledge modules into its knowledge database as are necessary to provide the monitoring services required to monitor the events for the console. The agent continually monitors the state of these registered events and sends messages to the console when a particular event (that the console has registered for) has occurred. See Col. 7, lines 15-31.

Differences between the Cited Art and the Present Invention

Claim 8 recites:

8. A computer workstation for use in a computer network having at least one process manager, the workstation comprising:
 - at least one application or process;
 - a network communication facility;
 - a configuration service layer in communicating relationship with the at least one application or process and the network communications facility,

wherein the at least one application or process and the configuration service layer cooperate to *generate and issue*, through the network communication facility, *a registration service request to the at least one process manager upon opening of the at least one application or process* at the computer workstation.

Applicant submits that neither Joyce nor Bonnell teach or suggest, either individually or combined, Applicant's claimed *generating and issuing a registration service request to a process manager upon opening of an application or process*. Both Joyce and Bonnell teach monitoring events and reporting the events. However, neither Joyce nor Bonnell teach or suggest *generating and issuing a registration service request... upon opening an application or process*. As recognized by the Examiner, Joyce fails to provide any teaching or suggestion for registration service requests. Bonnell describes registrations but as they relate to registering to receive information about an event before the event occurs. See Bonnell, Col. 7, lines 14-31, Fig. 19 and Fig. 25. Claim 8, on the other hand, recites generating and issuing a registration request after an event (i.e., the opening of an application or process) occurs which is different than Bonnell which teaches registering to receive an event before it occurs not afterwards.

For reasons set forth above, Applicant respectfully urges that neither Joyce nor Bonnell render Applicant's claimed invention obvious under 35 U.S.C. §103.

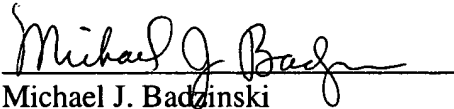
All independent claims are believed to be in condition for allowance. All dependent claims are dependent on believed to be allowable independent claims and are therefore believed to be in condition for allowance.

Quick favorable action is respectfully requested.

Please charge any additional fee occasioned by this paper to our Deposit Account

No. 03-1237.

Respectfully submitted,



Michael J. Badzinski

Reg. No. 51,425

CESARI AND MCKENNA, LLP

88 Black Falcon Avenue

Boston, MA 02210-2414

(617) 951-2500